

iOS 4 SDK入门——给JavaScript程序员 (影印版)

涵盖
iOS SDK 4.2



Learning the

iOS 4 SDK for JavaScript Programmers

O'REILLY®

东南大学出版社

Danny Goodman 著

iOS 4 SDK入门

——给JavaScript程序员（影印版）

Learning the iOS 4 SDK for
JavaScript Programmers

Danny Goodman

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

iOS 4 SDK 入门: 给 JavaScript 程序员: 英文 / (美)
古德曼 (Goodman, D.) 著. —影印本. —南京: 东南大
学出版社, 2011.5

书名原文: Learning the iOS 4 SDK for JavaScript Programmers
ISBN 978-7-5641-2710-7

I. ① i… II. ① 古… III. ① JAVA 语言—程序设
计—英文 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2011) 第 062876 号

江苏省版权局著作权合同登记
图字: 10-2010-288 号

©2010 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2011. Authorized reprint of the original English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2010。

英文影印版由东南大学出版社出版 2011。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

iOS 4 SDK 入门——给 JavaScript 程序员 (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 20 印张

字 数: 495 千字

版 次: 2011 年 5 月第 1 版

印 次: 2011 年 5 月第 1 次印刷

书 号: ISBN 978-7-5641-2710-7

定 价: 68.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

Preface

You don't have to be an Apple fanboy or fangirl to give Apple Inc. credit for redefining mobile gadgetry and its surrounding industries. First the company used the iPod to reshape the music industry and strongly influence how we acquire and consume tunes. Just count the number of people wearing iPod-connected earbuds in a subway car. Then the iPhone rewrote the cellular telephone industry manual, while opening the world's eyes to the potential of being connected to the Internet nearly everywhere, all the time. It's happening again with the iPad, where electronic publishing is evolving right before our eyes.

Although the iPhone was an early success with just the workable but limited set of Apple-supplied applications that came with the phone, programmers couldn't wait to get their hands on the platform. The first word that Apple let drop about third-party developers, however, landed with a bit of a thud: they were graciously allowed to create web apps. Sure, the iPhone's WebKit-based browser let creative HTML, CSS, and JavaScript programmers create far more than dull web pages, but the apps still faced frustrating limits compared to Apple's native apps.

It took some additional months, but Apple eventually released a genuine software development kit (SDK) to allow third-party programmers to create native applications for what was then called the iPhone OS. Part of Apple's task was also creating the App Store to distribute apps—yet another industry-transforming effort. Many existing Mac OS X developers rejoiced because the iPhone OS was derived from Mac OS X. The iPhone SDK was based on the same Xcode tools that Mac developers had been using for some time. The language of choice was Objective-C.

As a happy iPhone early adopter, I eagerly awaited the iPhone SDK. Unfortunately, despite my years of being a dedicated Mac user since 1984 and a scripter since 1987 and the HyperCard days, I had never done any Mac OS X programming. I didn't know much about C and next to nothing about Objective-C. Still, I thought perhaps my years of experience in JavaScript would be of some help. After all, at one time I even learned enough Java to write a small browser applet to demonstrate how JavaScript code in a web page can communicate with the applet. At least I knew what a compiler did.

When the iPhone SDK landed on my Mac, I was simply overwhelmed. The old metaphor of trying to sip from a firehose definitely applied. The more I read Apple's early developer documentation, the more I felt as though I had to know a lot more than I knew just to understand the "getting started" texts. With JavaScript having been the most recent language acquisition for me (albeit back in late 1995), I looked for anything I could borrow from that experience to apply to iPhone app development. I'd see occasional glimmers, but I was basically flying blind, not knowing what I had to discard and what I could keep.

The SDK was evolving during that time as well. I'd read a tutorial here and there, but I wasn't making much headway at first. Some tools, especially Interface Builder, felt incomplete to me. Frankly, I had a couple of false starts where I walked away until a future SDK version appeared. Finally, I reached a point that was "put up or shut up." After sticking with it and reading many of the documents many times, I was, indeed, getting tastes from the firehose. Working on iPhone development as a part-time effort over a three-month period, I managed to go from the starting line to submitting my first app to the App Store in January 2009.

Since then I've been monitoring the developer communities on both the native app and web app sides. I've even sat in online courses for web app developers to see what they're saying in the chat room. A lot of web app developers seem to look enviously to native iPhone and iPad development. I suspect many have gone through the same false starts that I did. And yet I know from my own experience that it is possible to make the transition from web app to native app developer if you know how to channel your JavaScript knowledge into what is now known as the iOS SDK environment.

What You Need to Start

I have written this book specifically for the web developer who is comfortable in the JavaScript language. Even if you use a bit of JavaScript to glue together apps from third-party JavaScript libraries and frameworks, you should be ready for this book. Unlike most entry-level iOS programming books, this one assumes that you have not necessarily worked in a compiled language before. You probably have little or no experience with C or Objective-C. But you do know what a string and an array are because you use them in your JavaScript work. I will be introducing you to the way Objective-C works by comparing and contrasting what you use in JavaScript. It's the kind of hand-holding that I wish I had when I started learning iPhone app development.

You will get more from this book if you are the adventurous type. By adventurous, I mean that you will follow the instructions throughout to try things for yourself. Along the way I will help you build an app called Workbench, where you will be able to play and learn by experimenting with little pieces of code here and there. Creating projects, editing files, and building apps is the only way to really get to know the SDK.

Of course, you'll need a Macintosh running Mac OS X version 10.6 (Snow Leopard) or later. I'll have more details about getting set up with hardware and SDK software in Chapter 2.

What's in This Book

Perhaps because my programming knowledge has been completely self-taught over the decades, this book does not follow what some might term traditional programming language training. First of all, you already come to the book with specialized knowledge. The goal of the book is to pick up where that knowledge leaves off and fill in the gaps with the new material. There's no doubt about it: there is a lot of new material for you. But I have tried to establish a learning progression that will make sense and keep you interested while you learn the decidedly unglamorous—but essential—parts of iOS programming.

Chapter 1 goes into detail about the differences between web app and native app programming for devices running iOS. It's not all roses for native app development, as you'll see, but I believe the positives outweigh the negatives. In Chapter 2, you will install the iOS SDK, inspect one of the sample apps, and run it on the iOS Simulator. Then in Chapter 3, I put you to work to create your first iPhone app—the Workbench app that you'll use throughout the rest of the book. The steps are intended to help you get more comfortable with Xcode and learn what it's like to work on an app in the environment.

In Chapter 4, you will use the Workbench app to build your first Objective-C object and compare the process against building the same object in JavaScript. You will spend a lot of time in Xcode. And if you've used JavaScript frameworks for your web app development, wait until you get a peek at the frameworks you'll be using in iOS app development.

The focus of Chapter 5 is understanding how the code you write commands an iOS device to launch your app and get it ready for a user to work with. In the process, you'll learn a great deal about how an app works. In fact, by the end of this chapter, you will add a second screen to Workbench and animatedly switch between the two.

Sometimes while learning new material, you have to take your medicine. That happens in Chapter 6, where you meet three programming concepts that are foreign to what you know from JavaScript: pointers, data typing, and memory management. There will be plenty of sample code for you to try in the Workbench app to learn these new concepts.

Objective-C is built atop the C language. There is still a bit of C that you should know to be more comfortable in the newer language. Chapter 7 shows you what you need to know from C. The good news is that a fair amount of it is identical to JavaScript. Hooray! And most of the esoterica isn't needed because it's all covered in more robust

and friendly ways in Objective-C, as covered in Chapter 8. There you'll learn how Objective-C handles strings, arrays, and other data collections.

The final chapter, Chapter 9, is also the longest. It provides a catalog of programming tasks you're accustomed to, but implemented in the iOS SDK. Most of the jobs will be familiar to you—formatting numbers, performing date calculations, sorting arrays, working with user-entered text, having Ajax-like communications with a server, and even dragging an item around a screen. I don't expect you to learn and remember everything described in Chapter 9, but know what's there and how to find it when the need arises in your own iOS development.

Two appendixes round out the offering. One provides tips on using the iOS SDK's documentation to its fullest extent. The other presents a list of common Xcode compiler errors that beginners encounter and what the errors really mean. Unintelligible error messages in the early going of learning a new environment can be very frustrating and discouraging. Appendix B makes it possible to learn more quickly from newbie mistakes.

Conventions Used in This Book

The following typographical conventions are used in this book:

Plain text

Indicates menu titles, menu options, menu buttons, and keys.

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, and directories.

Constant width

Indicates variables, methods, types, classes, properties, parameters, values, objects, XML tags, the contents of files, and logging output.

Constant width bold

Highlights new code or code of special importance in examples.

Constant width italic

Shows text that should be replaced with user-supplied values.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Learning the iOS 4 SDK for JavaScript Programmers* by Danny Goodman (O'Reilly). Copyright 2011 Danny Goodman, 9781449388454."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/9781449388454>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://www.oreilly.com>

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com>.

Acknowledgments

Having published over 45 books since the early 1980s, I have witnessed many changes across the computer-book universe. But one beacon of quality has always burned brightly: O'Reilly. The opportunity to publish a title through O'Reilly inspires an author to produce a work commensurate with an impeccable publishing record. It was a comfort to have super-knowledgeable editors Brian Jepson and Andy Oram challenging me to compose a better book at every step. Technical reviewers Alasdair Allan and Zachary Kessin responded above and beyond the call of duty to make sure my facts were factual and the reader's best interests were being served.

About the Author

Danny Goodman has been writing about personal computers and consumer electronics since the late 1970s. A freelance writer and programmer, he's published hundreds of magazine articles, several commercial software products, and three dozen computer books. His most popular book titles—on HyperCard, AppleScript, and JavaScript—have covered programming environments that are both accessible to non-professionals yet powerful enough to engage experts. He is currently an independent iOS app developer, with three products available on the App Store and more in the pipeline.

Colophon

The dog on the cover of *Learning the iOS 4 SDK for JavaScript Programmers* is a King Charles Spaniel. Today's Cavalier King Charles Spaniel is descended from a small, "toy" type of spaniel that was popular in 16th-century England. King Charles II, from whom the breed gets its name, was so fond of these dogs that he decreed that they were to be allowed in any public place, and it was said that "His Majesty was seldom seen without his little dogs." These spaniels were often referred to as "Comforters"; in the winter, a noble lady riding in a carriage was likely to keep a spaniel in her lap for warmth. While used by some for hunting small game, the King Charles Spaniel was typically valued for its companionship and considered more of a luxury item than a utilitarian pet.

Today's King Charles Spaniel emerged in part from interbreeding with the pug—which was in fashion in England during the reign of King William III and Queen Mary II—and the longer-nosed spaniels Charles II was so fond of. Their pointed noses, flat heads, and almond-shaped eyes were replaced with the shorter muzzles, domed skulls, and large, round eyes that characterize them today. The turn of the 20th century saw a final attempt to revive the breed as it existed during King Charles's time, but the modern King Charles Spaniel—named "Cavalier King Charles Spaniel" by the Cavalier Club in 1928—persisted. During World War II, the breed declined significantly (with one registered kennel dropping from 60 to 3 Cavaliers), but regained popularity after the war and throughout the 1940s.

Today, the Cavalier King Charles Spaniel is gaining popularity worldwide. There are national Cavalier breed clubs in about a dozen countries, including Finland, Italy, New Zealand, and South Africa. The Kennel Club reports that the Cavalier was the sixth most popular dog in the UK in 2007, and according to statistics from the American Kennel Club, they were the 25th most popular in the US in 2008, particularly in San Francisco, New York City, Boston, and Washington, D.C..

The cover image is from Wood's *Animate Creation*, Vol. I. The cover font is Adobe ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSansMonoCondensed.

O'Reilly Media, Inc. 介绍

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc. 达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书,包括:

- 《Hadoop 权威指南》(影印版)
- 《深入浅出 Python》(影印版)
- 《基于开源工具的数据分析》(影印版)
- 《界面设计模式,第 2 版》(影印版)
- 《挖掘社交网络》(影印版)
- 《Python 编程,第 4 版》(影印版)
- 《iOS 4 编程 Cookbook》(影印版)
- 《iOS SDK 入门——给 JavaScript 程序员》(影印版)
- 《Tapworthy: 设计伟大的 iPhone 应用》(影印版)
- 《网站运维》(影印版)
- 《Google App Engine 编程》(影印版)
- 《开发大规模 Web 应用》(影印版)
- 《Perl Cookbook, 第 2 版》(影印版)

Table of Contents

Preface	xi
 1. Why Go Native?	 1
Using an App Offline	2
More Access to the Hardware	3
More Access to the Software	4
What You Lose	6
Distribution	7
Apple iOS Developer Program	8
Content	8
Authoring Platform Choices	8
Taking the Plunge	9
 2. Welcome to the iOS SDK	 11
Hardware and OS Requirements	11
Installing the SDK	12
About iOS Developer Programs	12
Inside the SDK	14
Viewing Developer Documentation	15
Loading Code Samples	18
Setting the Project's Base SDK	21
Trying the iOS Simulator	22
Coming Up... ..	24
 3. Creating a Test Workbench	 25
Creating the Project in Xcode	26
Selecting a Project Type	26
Naming and Saving the New Project	29
Welcome to Your Project	29
Editing Your First Files	31
What the runMyCode: Method Does	34

Building the User Interface	35
Adding a Button to the View	38
Connecting the Button	42
Going for a Test Ride	46
Congratulations	49
4. Structural Overview of an iOS App	51
Where It All Begins: APIs	51
APIs You Already Know	51
The Cocoa Touch APIs	52
Frameworks	53
Foundation Framework	54
UIKit Framework	54
CoreGraphics Framework	55
Adding Frameworks	55
Frameworks Set in Stone	56
Welcome to Class Files	57
The JavaScript Way	57
The Objective-C Way	58
Header File Details	61
Using Xcode to Create DGCar Class Files	65
Editing the @interface Section	68
Message Passing	70
Editing the @implementation Section	70
Integrating the DGCar Class into Workbench	75
Creating Object Instances	76
NSLog() and String Formats	77
Running the Code	78
What About Accessing Instance Variables?	79
Recap	81
5. App Execution Flow	83
Some C Language Roots in an iOS App	83
An Introduction to Delegates	85
How UIApplication Appoints Its Delegate	87
The App's Info.plist File	87
Inside MainWindow.xib	88
iPhone App Development Design Patterns	92
The Model-View-Controller Design Pattern	92
Other Design Patterns	94
The Importance of Views	95
The App Window—UIWindow	96
Adding Another View to Workbench	97

Recap	106
-------	-----

6. Central Objective-C Concepts: Pointers, Data Types, and Memory Management ..

107	
Pointers	108
Pointers and Memory	108
Pointers and Objective-C Variables	110
Pointer Notation	111
Determining Pointer Usage	113
Data Typing	115
Objective-C Data Types	116
Cocoa Touch Data Types	116
Objective-C Variable Declarations	118
Objective-C Method Declarations	118
The id Data Type	122
Converting Objective-C Data Types	123
Memory Management	125
Cleaning Up After Yourself	125
The Retain Count	127
Autorelease Pools	129
Observing Memory Usage	130
Recap	131

7. C Language Fundamentals 133

Variable Names	133
Variable Scope	136
Instance Variables	137
Local Variables	137
Local Variables in Control Structure Blocks	138
Static Local Variables	140
Global Variables	140
Constant Values	141
Functions	142
C Structures	148
C Arrays	151
Enumerated Types	152
Operators	153
Program Flow Constructions	153
Boolean Values	154
Math Object Equivalents in C	155
Inserting Comments	157
Recap	157

8. Objective-C/Cocoa Touch Fundamentals	159
More About Classes	159
Temporary Objects	160
Subclassing Framework Classes	161
Defining Your Own Custom Subclasses	162
Adding to a Class Without Subclassing—Categories	166
Real Classes in Real Action	168
TheElements Overview	168
TheElements Class File Structure	171
Class Properties	175
Specifying Properties in the Header File	176
Synthesizing Properties in the Implementation File	178
Using Properties	178
Properties in Framework Classes	180
About NSString	181
Creating an NSString	182
JavaScript String Method Equivalents in Objective-C	185
NSMutableString	189
About NSArray	190
Creating an NSArray	192
Retrieving Array Elements	193
JavaScript Array Method Equivalents in Objective-C	193
NSMutableArray	194
About NSDictionary	195
Creating an NSDictionary	195
Retrieving Dictionary Entries	197
NSMutableDictionary	198
Arrays and Dictionaries in Action	199
Recap	201
 9. Common JavaScript Tasks in Cocoa Touch	 203
Formatting Numbers for Display	203
Preformatted Number Styles	204
Rounding Numbers for Display	206
Creating a Date Object	207
Adding a UIDatePicker to Workbench	207
Understanding NSDate	210
Creating a Date Object for a Specific Date	211
Extracting Components from an NSDate Object	213
Creating NSDate Objects from Strings	214
Converting an NSDate to a String	217
Calculating Dates	219
10 Days in the Future	219

Days Between Dates	220
Comparing Dates	221
Downloading Remote Files Asynchronously	222
Example Project	223
Creating the Request	224
Initializing the NSMutableData Object	225
Delegate Methods	226
Downloading Only When Needed	228
Accounting for Fast App Switching	231
Reading and Writing Local Files	233
iOS App Directories	233
Obtaining Directory Paths	235
Obtaining Paths to Files Delivered with Your App	236
Writing Files to Disk	236
Reading Files from Disk	238
Writing and Reading Property List Files	239
Performing File Management Tasks	240
Sorting Arrays	241
Sorting with a Selector	241
Sorting with a Function	243
Sorting Arrays of Dictionaries with NSSortDescriptor	245
Capturing User-Entered Text	246
The Code Portion	247
The Interface Builder Portion	250
Validating Text Entry with Regular Expressions	251
Modifying the Code	253
Modifying the User Interface	255
Using Regular Expressions for Text Search and Replace	255
Dragging a View Around the Screen	258
The Code Portion	259
The Interface Builder Portion	264
Recap	265
A. Getting the Most from Xcode Documentation	267
B. Common Beginner Xcode Compiler Errors	277
Glossary	281
Index	285